(51) International Patent Classification⁷: H04Q 7/32

(21) International Application Number: PCT/SE00/01015

(22) International Filing Date: 19 May 2000 (19.05.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
9901904-4   26 May 1999 (26.05.1999)   SE

(71) Applicant: TELEFONAKTIEBOLAGET LM ERICS-SON (publ) [SE/SE]; S-126 25 Stockholm (SE).

(72) Inventors: HALL, Hans; Hjulgatan 5, 2trp, S-754 20 Uppsala (SE). ERIKSSON, Michael; Barks Väg 12, 2trp, S-170 73 Solna (SE). FORSGREN, Gunnar; Gillevägen 1A, S-139 50 Värmdö (SE).

(74) Agent: BJELKSTAM, Peter; Bergenstråhle & Lindvall AB, P.O. Box 17704, S-118 93 Stockholm (SE).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
—   With international search report.
—   Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS OF DOWNLOADING INTO A RADIO TERMINAL
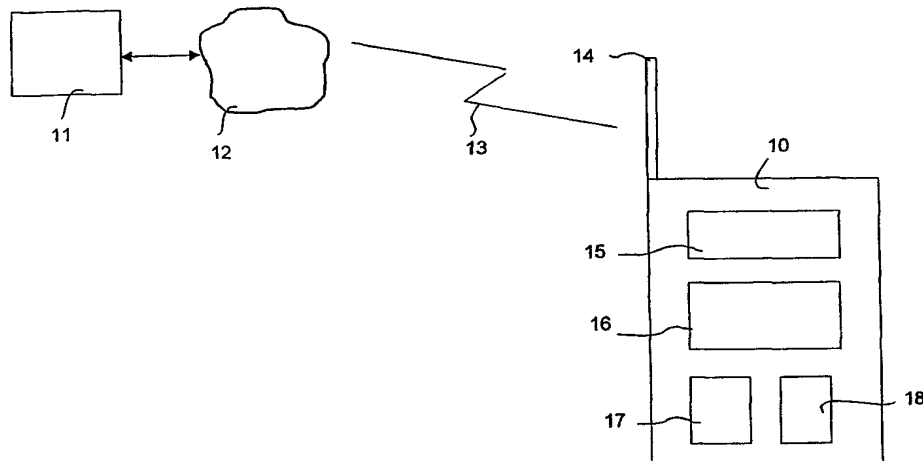
(57) Abstract: The invention is concerned with a method of downloading software into a radio terminal having at least two pieces of the software, one of which works as the current version. The capability of functioning of the current version of the software in the radio terminal is determined when the terminal is turned on. Another version of the software is downloaded if it is notified that there is a new version or if the current version does not work. If the current version is working, it is stored before a new version is downloaded. The capability of functioning of a current version of the software is tested before use. The result of the test is indicated in such a form that the capability of functioning of said tested version of the software can be determined. The invention is also concerned with a radio terminal comprising means to carry out the method of the invention.

METHOD AND APPARATUS OF DOWNLOADING INTO A RADIO TERMINAL

TECHNICAL FIELD

The invention is concerned with a method and apparatus of downloading software into a radio terminal, e.g. a mobile phone, and more particularly it is concerned with remotely upgrading of software.

DESCRIPTION·OF BACKGROUND ART

Radio terminals are typically programmed with two pieces of the software, a first piece is hard coded in a programmable read only memory (PROM) and a second, upgradable piece is loaded into flash Programmable Read Only Memory (flash-PROM). The software loaded in the flash-PROM may be periodically upgraded.

Usually, new versions of programs are made regularly and therefore there are methods and apparatuses to reprogram radio terminals, such as cellular telephones, remotely using a wireless communication link. It is advantageous if such methods retain the old software until the upgraded software has been tested and verified.

In the international patent application WO 98/38820 of the applicant, there is presented a method and radio apparatus for downloading software into a remotely located cellular telephone via wireless communication. The cellular telephone of this document includes two memories for storing software with one memory storing the current software and the second memory available for downloading new software. The cellular telephone also includes a controller for loading the received software into the cellular telephone and for performing a checksum on the new software to control the process for downloading the new software into the phone. If the calculated checksum does not match the transmitted checksum, a retransmission of the new software is carried out.

In some situations it is extremely important that the radio terminal, especially if it is a mobile phone, is functioning, e.g. in situations of emergency calls fort the user's

security. The software therefore has to be very robust. When the software is downloaded over a wireless or other network, and updated in the radio terminal, it is possible that a new version contains bugs and will perhaps not start properly or even not start at all. This might be the case when the software is written by a third party.

5   Moreover, the software could be quite advanced, so it could be hard to tell, from a programmer's view, when the application really has been started, even if a checksum process for the downloading process itself has been carried out. Even if a version is supposed to work, the program itself might have been corrupted in some way. The checksum only controls if the program sent from the server still is in the same form

10   when received by the radio terminal. If the program has bugs at the server, the same bugs are transferred to the radio terminal.

If a version of the software that has been installed appears not to work, there should be some method to obtain a working version. There is also a problem with knowing

15   when the application software has been started successfully.

SUMMARY OF THE INVENTION

20   One object of the invention is a method and apparatus with which the user always knows if the current software in the radio terminal is functioning.

Another object of the invention is a method and apparatus of downloading and installing software into the radio terminal in such a way that the above object is

25   fulfilled.

The method of the invention is therefore mainly characterized in that the capability of functioning of the current version of the software in the radio terminal is determined. Optionally, a possible existence of a new version of the software is then notified

30   before or after said determination. As a possible result of said determination and/or said notifying, another version is selected to be downloaded. If according to said determination, the current version of the software does not work, another version of the software will be downloaded. Before another version is downloaded, the current

version of the software is preferably stored if it is a working version according to said determination. The selected version of the software is downloaded into the radio terminal to be the current version if there according to said determination or said notifying is a reason thereto. The capability of functioning of the version of the

5   software, that now is the current version in the radio terminal, is then tested. As is indicated above, the current version might be the one which was in the terminal in the beginning, when the terminal was turned on, it can be another version or it can be a new version. The result of said test is indicated in such a form that the capability of functioning of said tested version of the software can be determined.

10

Thus, the invention also covers the methods, wherein the step of notifying the existence of a new version is excluded as well as methods wherein this step is a necessary step.

15   One version can alternatively also be tested more than once, before another version is installed.

The apparatus of the invention is mainly characterized by means for determination the capability of functioning of the current version of the software in the radio terminal,

20   notifying a possible existence of a new version of the software, selecting another version of software to be downloaded to replace the current version, storing the current version of the software, downloading another version of the software into the radio terminal to be the current version, testing the capability of functioning of the actual current version of the software, and of indicating the result of said test in such a

25   form that the capability of functioning of said tested version of the software can be determined.

In this application the term "downloading" means both remotely downloading and installing and reinstalling and reloading from another memory. The term "current

30   version" means the active version of software in the terminal. The other versions stored are inactive. It is pointed out that when a new version is available it is not necessary to download it immediately. If it is downloaded it is not necessary to start it immediately and it can be done later on, for example the next time the terminal is

turned on. Furthermore, the new version of the software can be a later version of the software, a better version or an improved version. The idea is that the possible existence of a "new" version is notified so that it would be possible to download some version wished which is not already stored in the terminal.

5

In the following there are presented some further, preferable characteristics for the method and apparatus of the invention.

When the radio terminal is turned on, the first thing that preferably is done is to
10   determine if the current version of the software is functioning. The determination can be done by identifying the presence or absence of an indicator that shows the software situation, and has been inserted in the terminal. The indicator can be the result of said test, which is indicated by the presence or absence of a special indicator that can be handled by an object in the program.

15

The sign can be a persistent object. If the object exists, it is known that the previous boot for downloading a new version of the software or the installing of another version failed. Such an embodiment is also possible, wherein the absence of the sign means that the current version does not work. A version that is supposed to work is
20   then installed and the method can be repeated immediately or later until the result of the test shows that the current version of the terminal is working. The method can also be repeated the next time the radio terminal is turned on, preferably each time the terminal is turned on.

25   Said other version of the software can be an earlier stored older version, preferably the foregoing version used. The number of versions to be stored is selected by deleting one or more of the older versions, preferably according as new versions are installed.

Said other version selected to be downloaded can be a new version of the software if
30   such a new version exist according to said notifying. The other version of the software selected to be downloaded is an earlier stored older version, if the current active version is not working according to said determination and if there is no new version or the new version is not wished to be installed.

If the persistent object exists when the system is booted up (or in the alternative embodiment, does not exist), it is a sign of that something went wrong the last time. As explained above, a working version is then installed to be the current version or the version is tested at least once more.

In one embodiment, said other version of the software to be installed as the current version can be a basic version stored in ROM or can be downloaded from the manufacturer's server if none of the stored older versions or the current version works. In some embodiments, there might be an option that the basic version can always be installed, if wished.

As indicated above, the result of said test is indicated by the presence or absence of a special indicator to show the software situation in the mobile phone by means of which indication the capability of functioning of the current version of the software in the mobile phone is notified.

When the capability of functioning of said installed/downloaded version of the software is presented by the absence of said special indicator and the opposite situation with the presence of said indicator, in one embodiment of the invention the indicator is removed, when according to the test, a working version of the software has been started. In another embodiment said indicator is updated every time the functionality test is carried out to show the number of tests performed or to identify the current version of the software.

The other version of software selected to be downloaded can be selected on the basis of the updated indicator. If the indicator for example shows that the test has been carried out many times, it may be advisable to reload a basic version of the software.

Thus, in one embodiment of the invention, instead of deleting the object when a test is passed and then creating a new one in the next test, the existing object is updated to give information of number of tests performed, of the actual version of the software etc. It can be decided in forward how many times a program shall be tested before

another or a basio version shall be installed. If the indicator shows up more than once,
it usually indicates that something else is wrong and in that case it might be preferable
to install the basic first version mentioned above already in this stage. Something
might for example have gone wrong when the program was stored in the memory. In
5   case of a third party code, another program might have destroyed the actual version.
The copying of files in the radio terminal is a further possible reason. It can therefore
also be advisable to test the same program more than once before installing an earlier
version or the basic version.

10   Said special indicator can be made in such a form that the actual version of the
software can be identified and it can be in form of an ordinary file.

The capability of functioning of said version of the software is tested by setting a
number of checkpoints the software should pass to start up correctly as a working
15   version. Each checkpoint can for example be a critical point per thread in the software
or per terminal resource in the software that has to be passed. A resource can be a
network, a file system, sound, etc. Other alternatives are also possible and they are
chosen depending on the program. A checkpoint counter can be used for counting the
checkpoints passed, the number of passed checkpoints can be checked, and the special
20   sign can be deleted or updated if the number of checkpoints in the counter is equal to
the number of checkpoints to be passed.

The current version can be stored by packing it with a compressing program to reduce
the amount of space required on the phone, for example the zip program.

25
The problem with knowing whether a new version of the software has successfully
started is thus solved in the invention by having an indicator, that either exists or not,
depending on the functionality of the software. The indicator will be deleted if the
new version has started or in an alternative embodiment, is retained if the new version
30   has started.

With the method and apparatus of the invention, one can always be sure to have a
working version and the system is very robust.

The radio terminal of the invention can be a computer or a mobile phone or the like.

In the following, the invention is described more in detail by means of four flow
5   schemes and a schematic view. It is pointed out that these descriptions only are
examples of embodiments of the invention and that the details of the inventions can
vary within the scope of the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

10
Figure 1 illustrates a schematic view of an apparatus for downloading software into a
radio terminal
Figure 2 is a general flow scheme of the invention.
Figure 3 is a flow scheme of two examples of a procedure in accordance with the
15   invention.
Figure 4 is a more detailed example of an embodiment of the invention.
Figure 5 is a detailed example of steps 5,8 and 9 of figure 2

DETAILED DESCRIPTION OF THE DRAWINGS

20
In figure 1, there is illustrated an example of an apparatus for remotely downloading
software into a radio terminal 10. An update server processor 11 communicates with a
network 12, which has a wireless communication link 13 to the radio terminal 10.
New versions of software are sent from the update server processor 11 via the network
25   12 and the wireless communication link 13 to the. radio terminal 10 through the
antenna 14. In addition to the normal functions of a radio terminal, the terminal 10 in
figure 1 contains a transceiver 15, a controller 16, a first memory 17 and a second
memory 18. The downloading of software from the update server 11 to the radio
terminal 10 is carried out through the antenna 14 via the controller 15 to one of the
30   memories.

When a new version of the software is available, the update server processor 11 can
transmit a message to the radio terminal 10 under control of controller 16, which can

either choose to download the new version or not in accordance with the invention. The availability of new versions of software can also be known for the radio terminal 10 so that the radio terminal 10 ask the update server 11 about the existence of a new version. The downloading itself is usually carried out by means of messages between

5   server processor 11 and radio terminal 10. The update processor 11 can also send older versions of the software to the radio terminal 10 in accordance with the invention.

When a new version has been downloaded in the terminal, it is considered as the

10   current version, the capability of functioning of which is tested in connection with the starting of the program. The starting process of the software is followed up through a number of checkpoints the software should pass in a correct start. A checkpoint counter can be used for counting the checkpoints passed, the number of passed checkpoints can be checked, and a special indicator can be created, deleted or updated

15   in dependence of the number of checkpoints in the counter being equal to the number of checkpoints to be passed.

In figure 2 there is presented a detailed embodiment of the invention by means of a flow scheme. One way of proceeding in the invention is by following the unbroken

20   arrows. Alternative ways of proceeding are indicated with dotted arrows. In some embodiments, the alternatives can be selected in accordance with predefined criteria.

Preferably, the method is carried out each time the terminal is turned on in accordance with reference number 1. In figure 2, the first thing to do is to check if the current

25   version of the software is a working version as indicated with step 2 in the figure. This determination, based on the checking, is in accordance with figure 1 carried out by means of the indication of a test result, the test having been carried out in accordance with step 8 and the indication in accordance with step 9 explained below. The changing of information between the indication of the test result of step 9 and the

30   checking of step 2 is indicated with a two-way arrow 3. The indication of the test result can be performed in many ways. It can be a sign in form of a persistent object that is an ordinary file.

If the current version, according to the test result indication of step 9, appears not to work, it means that something went wrong in the previous boot of the software. Step 4a can now be carried out, according to which another version that is supposed to work is installed. This version might be an earlier stored working version of the software or it can even be a basic version stored in ROM or it can even be downloaded from the server of the manufacturer. If the current version appears not to work, it can also be tested once more in steps 8 and 9. This alternative way is presented with a dotted arrow in the figure.

If the current version in accordance with the test result presentation of step 9 is a working version considered in step 2, then, it is in step 4b of figure 2 checked if there is a new version existing. The alternative of not checking for a new version is indicated with a dotted arrow, and in that case the current version is started as indicated in step 5. The capability of functioning of the current version is tested as indicated in step 8 when the software is started in accordance with checkpoints set in the program.

It is possible to download a new version of the software even if there is no earlier version in the radio terminal, but this embodiment is not illustrated. In that case, the first step in the invention is to download the new version, which then is started and tested and the test result is indicated as is described.

If there is no new version (or it is not downloaded), the current version is started as indicated in step 5, and tested, as indicated in step 8.

The alternative of not downloading the new version of the software is indicated with a dotted arrow.

If there is a new version, the current version is in figure 2 stored in step 6 and the new version is downloaded in step 7.

For the new version downloaded in step 7 a functionality test is performed in connection with starting up as indicated in steps 5 and 8.

The capability of functioning of the downloaded or installed version of the software is tested as indicated in step 8 by setting a number of checkpoints the software should pass to start up correctly as a working version. A checkpoint counter can be used for
5   counting the checkpoints passed, the number of passed checkpoints can be checked, and the special sign can be deleted or updated if the number of checkpoints in the counter is equal to the number of checkpoints to be passed.

The result of the test is indicated in step 9 in such a form that the capability of
10   functioning of a current version of the software always can be determined. Therefore, the result of said test is indicated by the presence or absence of a special indicator or sign to show the software situation in the terminal by means of which indication the capability of functioning of the current version of the software in the terminal is notified.
15

A new downloaded version or an installed version is tested in step 8 in connection with the starting of the program. There are, however, checkpoints embedded, and if these checkpoints are not passed, it shall appear in the presentation of the test result.

20   Figure 3 is a detailed flow scheme of two examples of how the method of the invention might proceed. It has to be pointed out that the downloading procedure depends on how the software in the terminal works and on the availability of new versions why the example is not intended to restrict the invention in anyway, but is shown for illustrating purposes only.
25

The terminal is turned on in accordance with reference number 100. In this embodiment, the indication of the test result is performed in form of a persistent object, the existence of which shows that the current version of the software in the terminal is not working.
30

In figure 3, step 200 shows in the first example shown in figure 3 that the object exists, which means that something went wrong in the previous boot of the software. Step 400 a1 can now be carried out, according to which another version that is

supposed to work is installed. This version might be an earlier stored working version 19.05.20 of the software or it can even be a basic version stored in ROM or it can even be downloaded from the server of the manufacturer. The downloaded version is started in step 500 whereby a new persistent object is created. In step 800:2, the downloaded
5   version is tested and, since the test is passed, the persistent object is deleted in step 900:2.

Returning now to step 200, the absence of a persistent object in the second example causes the program to execute step 400 b, whereby it is now checked if there is a new
10   version existing. Since there is a new version, the current version is stored in step 600 and the new version is stored in step 700 to be the current version. The new version is started in step 800:1 hereby a persistent object is created and a number of checkpoints to be passed is set. In step 900:1, the new version is tested. As this version did not start up correctly, actions are taken in step 400:a2 to remedy the fault by for example
15   by going to step 400a1 and by proceeding as was explained in the first example.

Other alternatives are discussed in the following. For example, the user of the terminal may conclude, from an improper program behaviour, that something is wrong and restart the terminal by first turning it off and immediately thereafter turning it on again
20   which effectively means to turn to step 100. It is also possible that the program execution is terminated and control is handed over to the kernel whereby the number of passed checkpoints may be checked. Automatic actions may then be executed in response to a mismatch between the preset counter value and the number of passed checkpoints. It is also possible to implement a low level interrupt in step 800, whereby
25   the kernel may detect that the program execution has stopped or entered into a loop. The kernel may then, in step 400:a2, execute automatic actions to remedy the fault, e.g. restart the program execution from step 200. If the kernel can take control, the automatic actions in case of a faulty program can be varied. For example the kernel may indicate to the user that a program error has been detected but that certain
30   functions will be allowed. The user may, e.g., be allowed to manage a local calendar whereas a network connection may not be allowed. The kernel may determine which functions of the program that are working from an analysis of data compiled from the checkpoints having been passed.

In figure 4, there is shown a detailed example of an embodiment of the invention.
In the figure, the different steps corresponding to the steps of figure 2 have been
indicated with corresponding reference numbers as in figure 2, except that in this

5 embodiment, the upper indexes ' and '' and ''' is used to indicate this special
embodiment. Lower indexes a, b, c have been used to indicate alternative ways. The
different steps in themselves have been earlier explained in the text, and the intention
with this figure is to show an example of an implementation of the invention.

10 The terminal is turned on in step 1'. The existence of a new version is notified in step
4b'. If there is a new version, the current version is moved in step 6'. (If the current
version is a working version it is stored, otherwise it is deleted). The new version is
loaded in step 7' and started in step 5a'. In connection with the starting there is created
an object in step 9b' to indicate passing of the test performed in step 8'. The passing

15 of the test is notified in step 2b'. A restart according to step 1b' is carried out if the
test is not passed. In that case the capability of functioning of the current version is
notified in step 2'. In this embodiment, if an object exist, it is notified in step 2'' if
this is the second try of booting the software. (This can be notified by updating the
object in such a way in every test not being passed). In figure 4, a basic version is

20 reloaded in step 4a', if in step 2'', it was notified that this was the second try of
booting.

If step 2'' showed that this was not the second try, it is notified in step 2''', if an old
current version exists e.g. as indicated by the second memory 18 in figure 1 being

25 loaded. This can be notified for example with creating an own object for this purpose.
If this is the case, the old version is reloaded in step 4'' and started and tested in step
5c' also comprising the creation of a new object. The passing of the test is notified in
step 2c'. If the test is passed, the object is deleted in step 9d' and the current software
version can be used.

30
If it was notified in step 2c', that the test was not passed, the current object is updated
in step 9e' to indicate in a subsequent execution of step 2'', a second try. A restart,
manual or automatical, is thereafter carried out in step 1b'.

If according to step 2''', there is no older working version, the basic version is
reloaded in step 4a'. Program execution then proceeds from steps 5b', 9b' by testing
the reloaded basic version.

5

If according to step 2', there is no object, it means that the current version is working
and it is started in steps 5b'. An object is created in step 9a', which is deleted in step
9c' if the test according to step 8a' is passed, after which the program is ready for use.

10   If the test is not passed in step 2a', a restart is carried out in step 1b'.

If there is no new version of software available according to step 4b', the next step is
2', which is explained above.

15   Figure 5 is a detailed example of a functionality test performed for the current
software in the terminal in connection with the starting in step 5. An object 52 is
created in step 51A. The functionality of the version of the software is tested by
setting a number of checkpoints C1, C2, C3 called checkpoint 1, checkpoint 2 and
checkpoint 3 in figure 5, the software should pass to start up correctly as a working

20   version. The number of checkpoints to be passed is defined in step 51B and stored in
step 53 in the object 52. The number of checkpoints can vary, but for illustrative
purposes, there are three indicated in figure 5. The number of checkpoints is contained
in the program and copied to the object 52 at an early phase of program execution.

25   Each checkpoint can for example be a critical point per thread in the software that has
to be passed. In figure 5, only one thread is described. Other alternatives are also
possible and they are chosen depending on the program. A checkpoint counter 54 can
be used in the object for counting the checkpoints passed, and each time a checkpoint
is passed, the counter number is increased by one. In figure 5, there has been

30   illustrated the case, wherein all three checkpoints has been passed.

The number of checkpoints passed is compared to the defined amount in step 55, and
if the number of checkpoints defined in the object is equal to the number of

checkpoints passed according to the counter, the test is passed as indicated by step 56 19.052

and the software is ready for use. If the number defined in the object is not equal to

the number of checkpoints passed, in this embodiment, the old object is updated in

step 57 or a new object is created.

CLAIMS

1. A method of downloading software into a radio terminal having at least two pieces of the software, one of which works as the current version,

   c h a r a c t e r i z e d by the steps of

   a) determining the capability of functioning of the current version of the software in the radio terminal,

   b) optionally, notifying a possible existence of a new version of the software,

   c) selecting another version of the software to be downloaded as a possible result of step a) or b),

   d) possibly storing the current version of the software, before a new version is downloaded,

   e) downloading of the other version of the software selected in step c) into the radio terminal to be the current version,

   f) testing the capability of functioning of the current version of the software of step a) or step e), and

   g) indicating the result of the test of step f) in such a form that the capability of functioning of said tested version of the software can be determined.

2. A method of downloading software into a radio terminal having at least two pieces of the software, one of which works as the current version,

   c h a r a c t e r i z e d by the steps of

   a) determining the capability of functioning of the current version of the software in the radio terminal,

   b) notifying a possible existence of a new version of the software,

   c) selecting another version of the software to be downloaded if the current version does not work or if there is a new version of the software available,

   d) storing the current version of the software, before a new version is downloaded, if the current version is a working version,

   e) downloading of the other version of the software selected in step c) into the radio terminal to be the current version,

   f) testing the capability of functioning of the current version of the software of step a) or step e), and

g) indicating the result of the test of step f) in such a form that the capability of functioning of said tested version of the software can be determined.

3. A method of downloading software into a radio terminal having at least two pieces of the software, one of which works as the current version, c h a r a c t e r i z e d by the steps of

a) determining the capability of functioning of the current version of the software in the radio terminal,

c) selecting another version of the software to be downloaded if the current version does not work,

e) downloading of the other version of the software selected in step b) into the radio terminal to be the current version,

f) testing the capability of functioning of the current version of the software of step a) or step c), and

g) indicating the result of the test of step d) in such a form that the capability of functioning of said tested version of the software can be determined.

4. Method of claim 1, c h a r a c t e r i z e d in that in step d), the current version is stored before a new version is downloaded if the current version according to step a) is a working version.

5. Method of claim 1, c h a r a c t e r i z e d by the steps of

a) determining the capability of functioning of the current version of the software in the radio terminal,

c) selecting another version of the software to be downloaded if the current version is not a working version according to step a),

e) downloading of the other version of the software selected in step c) into the radio terminal to be the current version,

f) testing the capability of functioning of the current version of the software of step e), and

g) indicating the result of said test in such a form that the capability of functioning of said tested version of the software can be determined.

6. Method of claim 1, c h a r a c t e r i z e d by the steps of

a) determining the capability of functioning of the current version of the software in the radio terminal,

b) notifying a possible existence of a new version of the software, before or after step a),

c) selecting another version of the software to be downloaded as a possible result of step a) or b),

d) storing the current version of the software, before a new version is downloaded, if the current version according to step a) is a working version,

e) downloading of the other version of the software selected in step c) into the radio terminal to be the current version,

f) after step a), or e), testing the capability of functioning of the current version of the software of step a) or step e), and

g) indicating the result of said test in such a form that the capability of functioning of said tested version of the software can be determined.

7. Method of any of claims 1 - 6, c h a r a c t e r i z e d in that the method is repeated until there is a working version of the software in the radio terminal.

8. Method of any of claims 1 - 7, c h a r a c t e r i z e d in that the capability of functioning of a current version is tested and determined at least once.

9. Method of any of claims 1 – 8, c h a r a c t e r i z e d in that steps a) - g) are carried out each time the mobile phone is turned on.

10. Method of any of claims 1 - 9, c h a r a c t e r i z e d in that the number of versions stored in the radio terminal is selected by deleting one or more of the older versions.

11. Method of claim 10, c h a r a c t e r i z e d in that the number of versions stored in the radio terminal is selected by deleting one or more of the older versions. according as new versions are installed.

12. Method of any of claims 1 – 2, 4,6 -11, c h a r a c t e r i z e d in that in step c), the other version selected to be downloaded is a new version of the software if such a new version exist according to step b).

13. Method of any of claims 1 – 11, c h a r a c t e r i z e d in that in step c) the other version of the software selected to be downloaded is an earlier stored older version, if according to step a), the current version is not working.

14. Method of any of claims 1 – 11, c h a r a c t e r i z e d in that in step c) the other version of the software selected to be downloaded is an earlier stored older version, preferably the foregoing working version stored, if according to step a), the current version is not working and according to step b), there is no new version available.

15. Method of any of claims 1 – 11, c h a r a c t e r i z e d in that in step c), the other version of the software selected to be downloaded is a basic version stored in ROM if that is the foregoing stored working version or if the method is repeated until that is the only working version of the software in the radio terminal.

16. Method of any of claims 1 – 11, c h a r a c t e r i z e d in that in step c), the other version of the software selected to be downloaded is downloaded from the manufacturer's server.

17. Method of any of claims 1 - 16, c h a r a c e r i z e d in that in step g), the result of said test is indicated by the presence or absence of a special indicator to show the software situation in the mobile phone by means of which indicator the capability of functioning of the current version of the software in the mobile phone is notified in step a).

18. Method of claim 17, c h a r a c t e r i z e d in that in step g), the capability of functioning of said current version of the software is indicated by the absence of said special indicator and the opposite situation with the presence of said

indicator, which is removed when, according to the test, a working version~~ of~~
software has been started.

19. Method of claim 17, c h a r a c t e r i z e d in that in step g), the capability of
    functioing of said current version of the software is indicated by the absence of
    said special indicator and the opposite situation with the presence of said
    indicator, which is updated every time step g) is carried out to show the number of
    tests performed.

20. Method of claim 19, c h a r a c t e r i z e d in that the other version of software
    selected to be downloaded is selected in step c) on the basis of the updated
    indicator.

21. Method of claim 17, c h a r a c t e r i z e d in that in step g), the capability of
    functioning of said current version of the software is indicated by the absence of
    said special indicator and the opposite situation with the presence of said
    indicator, which is updated every time step g) is carried out so that the current
    version of the software can be identified.

22. Method of any of claims 17 - 20, c h a r a c t e r i z e d in that said special
    indicator is made in such a form that the current version of the software can be
    identified.

23. Method of any of claims 17 – 22, c h a r a c t e r i z e d in that said indicator is in
    form of an ordinary file.

24. Method of any of claims 1 – 23, c h a r a c t e r i z e d in that in step f), the
    capability of functioning of said current version of the software is tested by setting
    a number of checkpoints the software should pass to start up correctly as a
    working version.

25. Method of claim 24, c h a r a c t e r i z e d in that each checkpoint is made in form of one critical point per thread in the software or per terminal resource in the software that has to be passed.

5   26. Method of claim 24 or 25, c h a r a c t e r i z e d in that in the testing of step f), the following steps are included

notifying each checkpoint passed with a checkpoint counter,

checking the number of checkpoints, and

indicating the tested version of software as a working version in step g) if the

10   number of checkpoints in the checkpoint counter correspond to the number of checkpoints to be passed.

27. Method of any of claims 1 – 26, c h a r a c t e r i z e d in that in step d), the current version is stored by packing it with a compressing program to reduce the

15   amount of space required on the phone.

28. Method of any of claims 1 – 27, c h a r a c t e r i z e d in that said radio terminal is a mobile phone.

20   29. Radio terminal having at least two pieces of software, one of which works as the current version, c h a r a c t e r i z e d by means for

a) checking the capability of functioning of the current version of the software in the radio terminal,

b) notifying a possible existence of a new version of the software,

25   c) selecting another version of software to be downloaded to replace the current version,

d) storing the current version of the software,

e) downloading another version of the software into the radio terminal to be the current version,

30   f) testing the capability of functioning of the current version of the software, and of

g) indicating the result of said test in such a form that the capability of functioning of said tested version of the software can be checked.

30. Radio terminal of claim 29, c h a r a c t e r i z e d in that there is one or more working versions stored.

5   31. Radio terminal of claim 30, c h a r a c t e r i z e d in that one of the versions of the software stored is a basic version stored in ROM.

32. Radio terminal of any of claims 29 - 31, c h a r a c e r i z e d by means for indicating the result of said test by the presence or absence of a special indicator to

10   show the software situation in the radio terminal.

33. Radio terminal of claim 32, c h a r a c t e r i z e d by means for updating the indicator every time said test is repeated to show the number of tests performed.

15   34. Radio terminal of claim 32, c h a r a c t e r i z e d by means for creating said special indicator in such a form that the current version of the software can be identified.

35. Radio terminal of any of claims 29 - 34, c h a r a c t e r i z e d by means for

20   creating said indicator in form of an ordinary file.

36. Radio terminal of any of claims 29 - 35, c h a r a c t e r i z e d by means for testing the current version of the software by setting a number of checkpoints the software should pass to start up correctly as a working version.

25

37. Radio terminal of claim 36, c h a r a c t e r i z e d by
a checkpoint counter notifying each checkpoint passed ,
means for checking the number of checkpoints, and
means for indicating the test result in accordance with the number of checkpoints

30   passed.

38. Radio terminal of any of claims 29 - 37, c h a r a c t e r i z e d by means for storing the current version by packing it with a compressing program to reduce the amount of space required on the phone.

5   39. Radio terminal of any of claims 29 - 38, c h a r a c t e r i z e d in that it is a mobile phone.
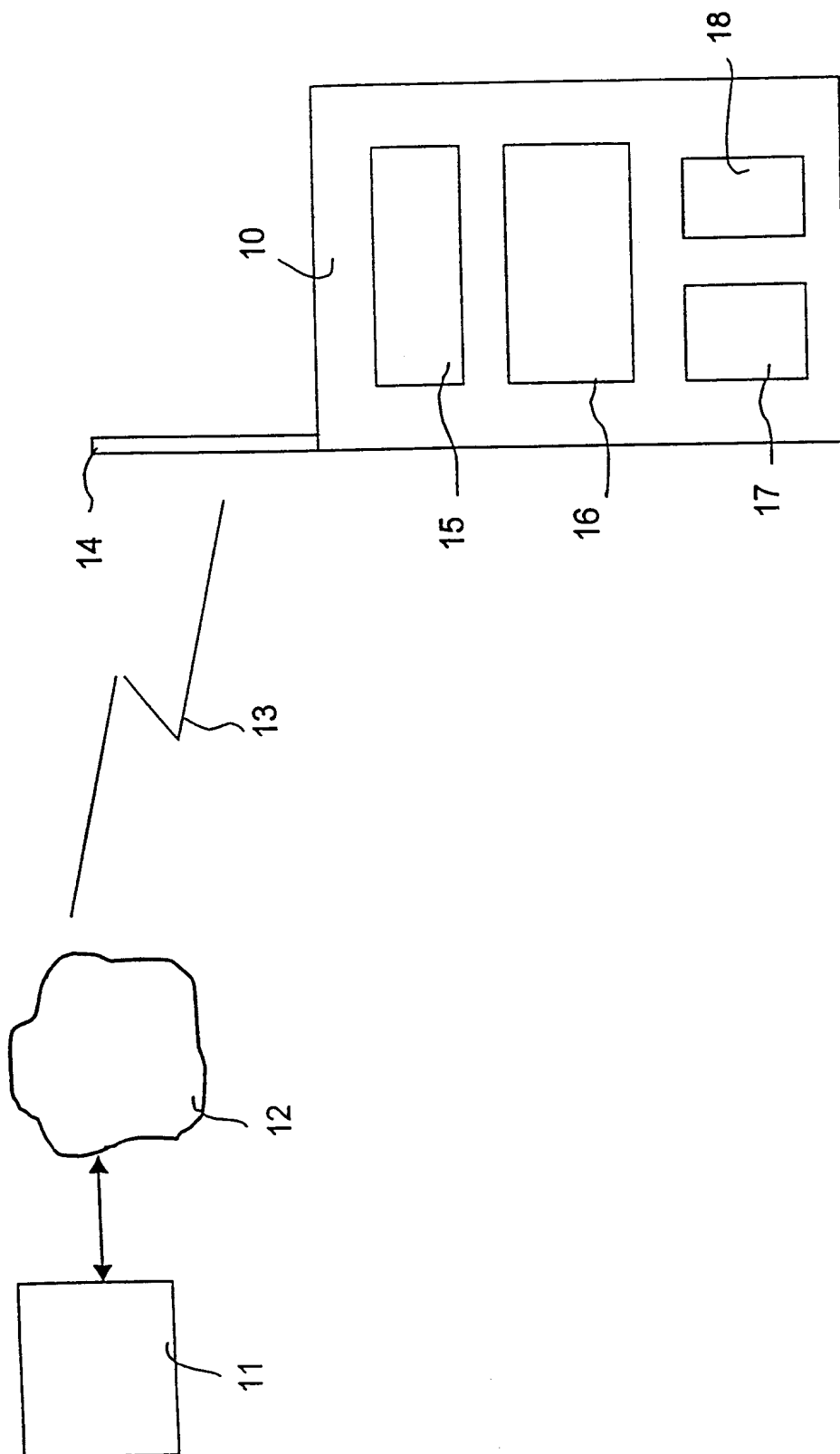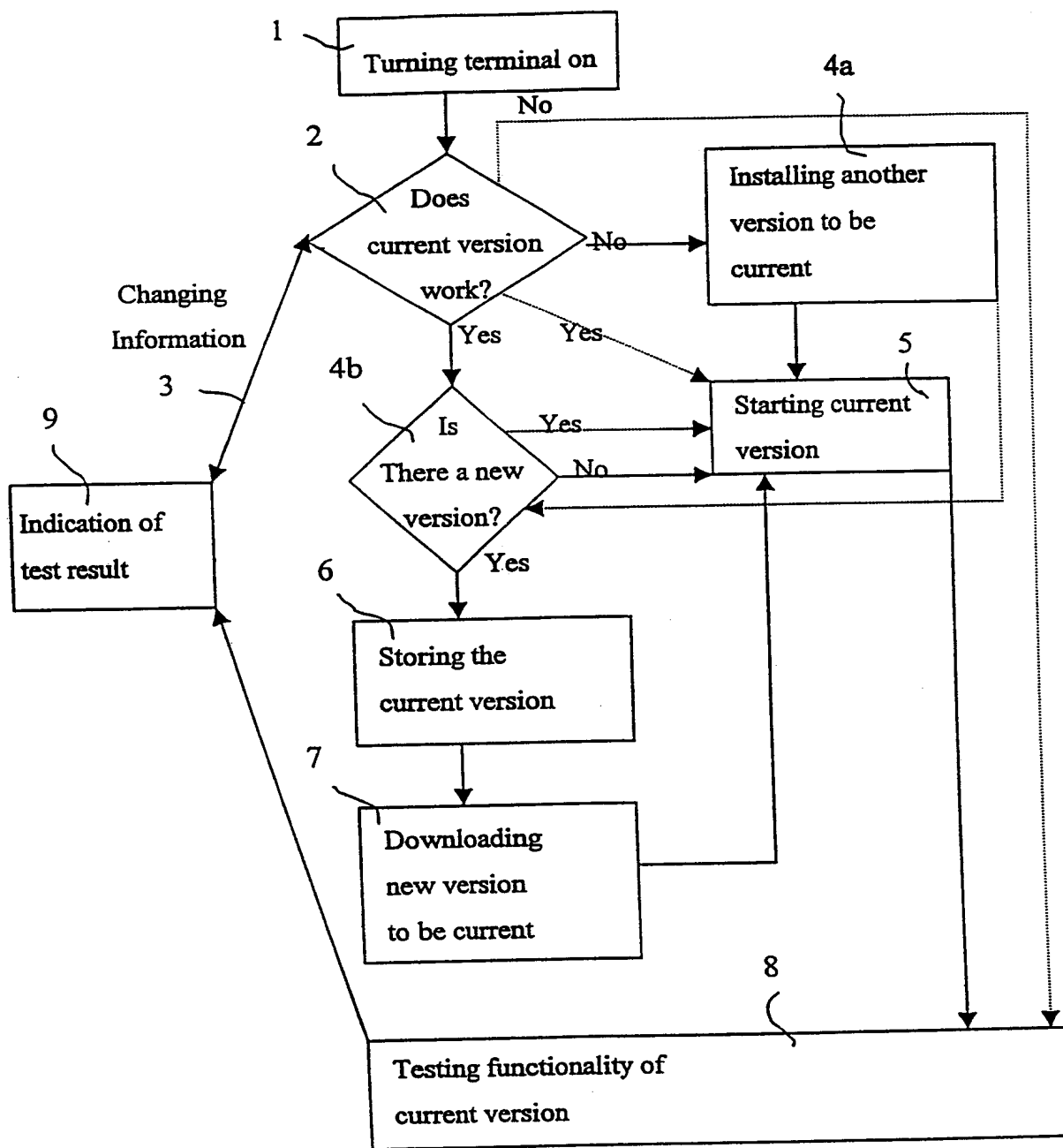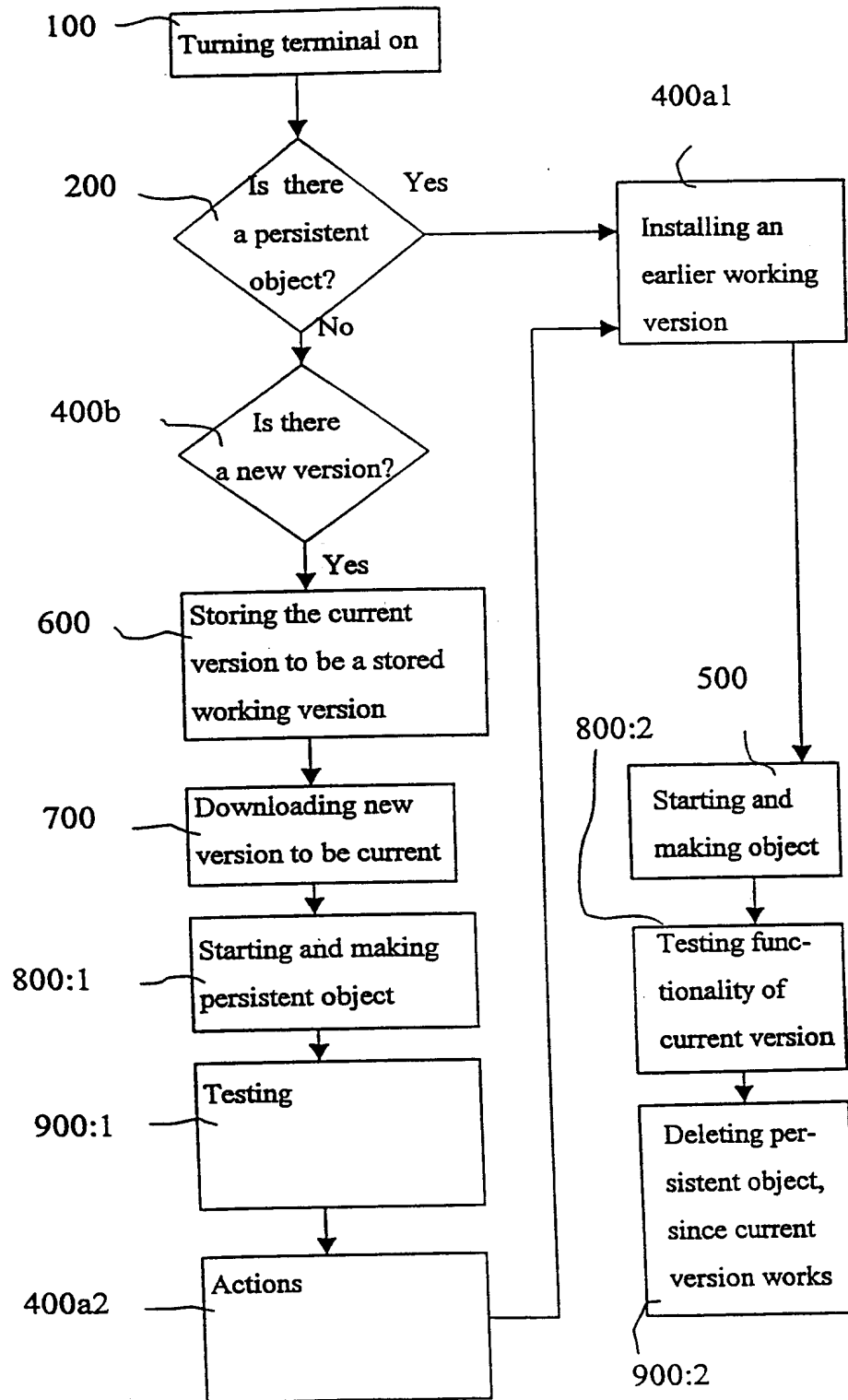
10

15

20

25

30

FIG. 1

FIG. 2

100 — Turning terminal on

200 — Is there a persistent object?

     Yes →

400a1 — Installing an earlier working version

No ↓

400b — Is there a new version?

Yes ↓

600 — Storing the current version to be a stored working version

700 — Downloading new version to be current

800:1 — Starting and making persistent object

900:1 — Testing

400a2 — Actions

500 — Starting and making object

800:2 — Testing functionality of current version

900:2 — Deleting persistent object, since current version works
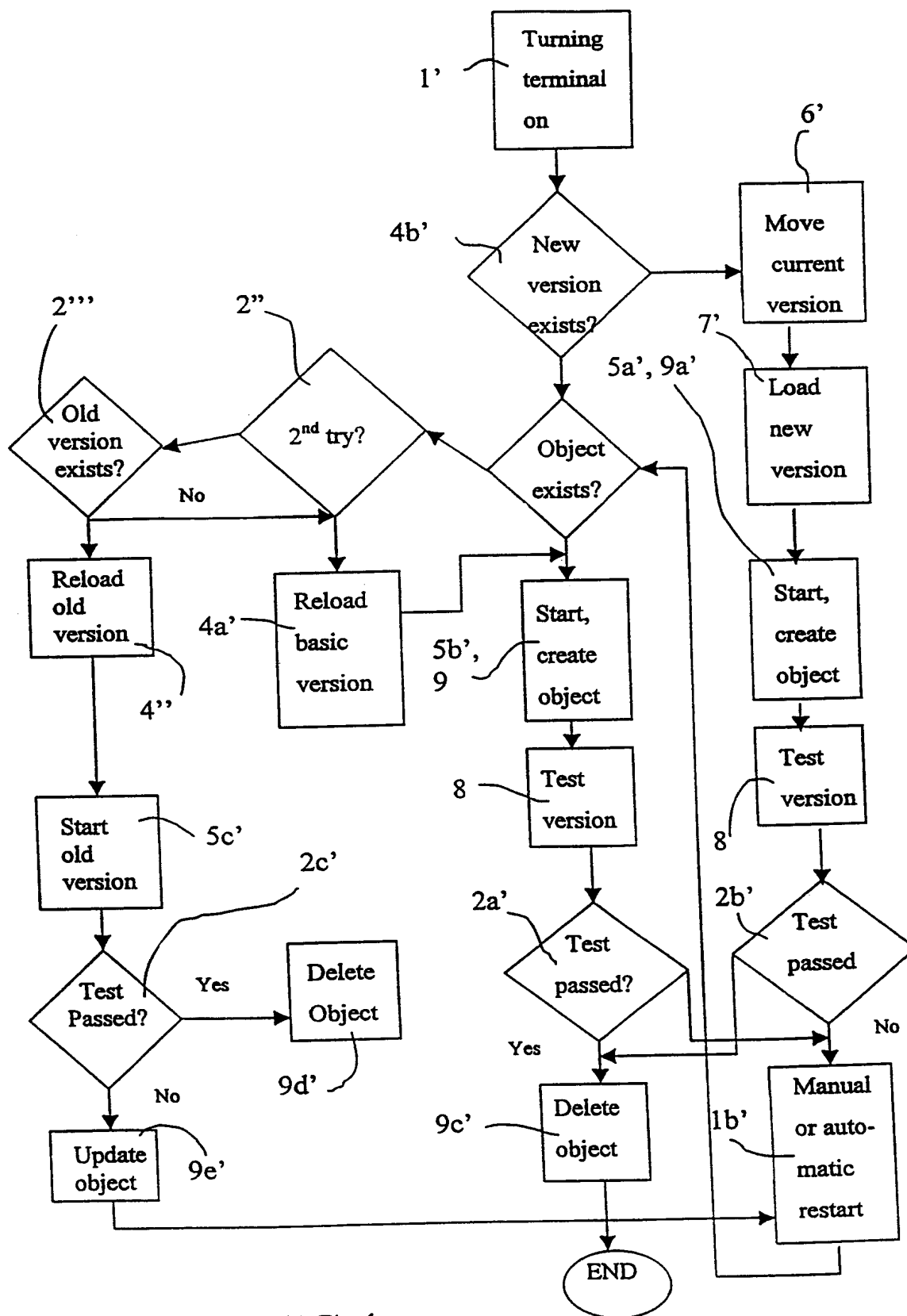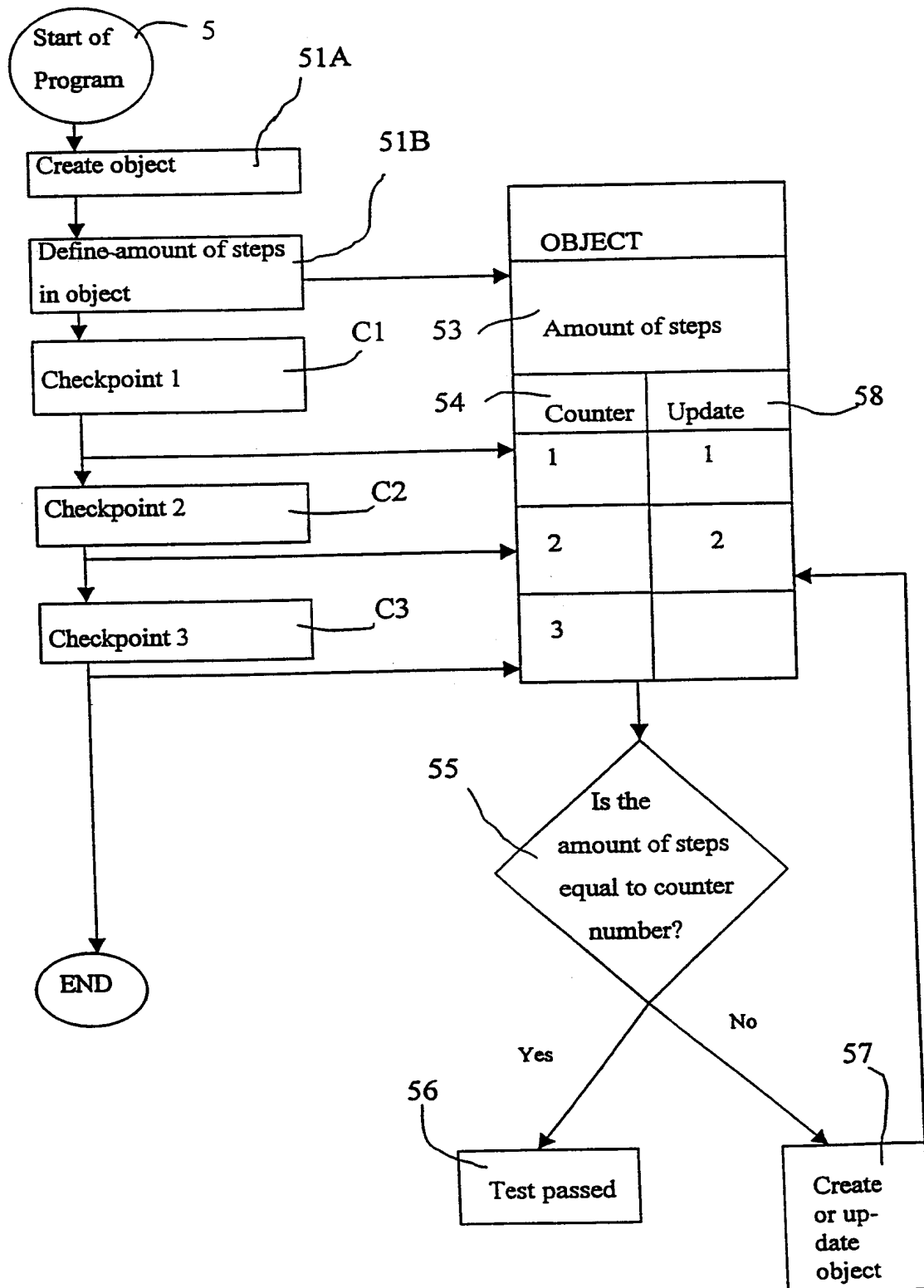
FIG. 3

FIG.4

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01015

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04Q 7/32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5155837 A (CHENG-CHUNG LIU ET AL), 13 October 1992 (13.10.92), column 2, line 32 - line 60; column 7, line 63 - column 8, line 22 --	1-39
A	US 5956480 A (Y.KURIHARA), 21 Sept 1999 (21.09.99), column 2, line 23 - column 3, line 2 --	1-39
A	WO 9716938 A1 (NOKIA TELECOMMUNICATIONS OY), 9 May 1997 (09.05.97), abstract --	1-39
A	EP 0767426 A1 (SIEMENS AKTIENGESELLSCHAFT), 9 April 1997 (09.04.97), abstract --	1-39

X	Further documents are listed in the continuation of Box C.		X	See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	erlier document but published on or after the international filing date	"X"	document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 October 2000	1 8 -10- 2000

Name and mailing address of the ISA/	Authorized officer
Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. + 46 8 666 02 86	Jerry Vennerholm/itw Telephone No. + 46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 00/01015

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	DE 19741703 A1 (SIEMENS AG), 1 April 1999 (01.04.99), abstract	1-39

--

International application No.
PCT/SE 00/01015

Patent document cited in search report			Publication date	Patent family member(s)			Publication date
US	5155837	A	13/10/92	CA	1326302	A	18/01/94
US	5956480	A	21/09/99	JP	7141277	A	02/06/95
WO	9716938	A1	09/05/97	AU	6877696	A	22/05/97
				EP	0872138	A	21/10/98
				FI	955188	A	24/06/97
				JP	11514760	T	14/12/99
EP	0767426	A1	09/04/97	NONE			
DE	19741703	A1	01/04/99	NONE			